

## Contents

<b>1</b>	<b>Routine/Function Prologues</b>	<b>2</b>
1.0.1	getpers.F90 (Source File: getpers.F90) . . . . .	2
1.0.2	persfile (Source File: getpers.F90) . . . . .	3

# 1 Routine/Function Prologues

## 1.0.1 *getpers.F90* (Source File: *getpers.F90*)

Opens and reads global precipitation forcing

CTIME = Current time

FTIMENRL = Nearest future data for NRL data

FTIMEHUFF = Nearest future data for HUFFMAN data

FTIMEPERS = Nearest future data for PERSIANN data

### REVISION HISTORY:

17 Jul 2001: Jon Gottschalck; Initial code

10 Oct 2001: Jon Gottschalck; Modified to adjust convective precip  
using a ratio of the model convective / total ratio

30 Jul 2002: Jon Gottschalck; Added PERSIANN and HUFFMAN global observed precip data sources

### INTERFACE:

```
subroutine getpers
```

#### *USES:*

```
use lisdrv_module, only : lis, gindex
```

```
use time_manager
```

```
use time_module, only : tick
```

```
use persdomain_module, only : persdrv
```

```
implicit none
```

### CONTENTS:

```
!-----
! Set parameter to measure 1.5 hour time offset when using HUFFMAN
!-----
gap = 0.0001712328767098370
!-----
! Determine required observed precip data times
! (current, accumulation end time)
! Model current time
!-----
yr1 = lis%t%yr !current time
mo1 = lis%t%mo
da1 = lis%t%da
hr1 = lis%t%hr
mn1 = lis%t%mn
ss1 = 0
ts1 = 0
call tick( ctime, doy1, gmt1, yr1, mo1, da1, hr1, mn1, ss1, ts1 )
!-----
```

```

! PERSIANN product end time
!-----
yr4 = lis%t%yr  !end accumulation time data
mo4 = lis%t%mo
da4 = lis%t%da
hr4 = 1*(lis%t%hr/1)
mn4 = 0
ss4 = 0
ts4 = 1*60*60
call tick( ftime_pers, doy4, gmt4, yr4, mo4, da4, hr4, mn4, ss4, ts4 )

!-----
! Ensure that data is found during first time step
!-----
if ( lis%f%gpcpsrc.eq.2 .and. get_nstep().eq. 1 ) endtime_pers = 1
!-----
! Check for and get Persiann Precipitation data
!-----
if (lis%f%gpcpsrc.eq.2) then
  if ( ctime > persdrv%perstime ) endtime_pers = 1
  if ( endtime_pers == 1 ) then !get new time2 data
    ferror_pers = 0
    call persfile( name, persdrv%persdir, yr4, mo4, da4, hr4 )
    print*, 'Getting new PERSIANN precip data',name
    call glbprecip_pers( name, ferror_pers )
    persdrv%perstime = ftime_pers
  endif !need new time2
endif
return

```

---

## 1.0.2 persfile (Source File: getpers.F90)

This subroutine puts together PERSIANN file name for 1 hour file intervals

### INTERFACE:

```
subroutine persfile( name, persdir, yr, mo, da, hr)
```

### CONTENTS:

```

92 format (80a1)
93 format (a80)
94 format (i4, i2, i2, i2)
95 format (10a1)
96 format (a40)
97 format (a8)

```

```

98 format (a1, i4, i2, a1)
99 format (8a1)
89 format (a7)
!-----
! Make variables for the time used to create the file
! We don't want these variables being passed ou
!-----
    uyr = yr
    umo = mo
    uda = da
    uhr = 1*(hr/1) !hour needs to be a multiple of 3 hours
    umn = 0
    uss = 0
    ts1 = -24*60*60 !one day interval to roll back date.

    open(unit=90, file='temp', form='formatted',access='direct', recl=80)
    write(90, 96, rec=1) persdir
    read(90, 92, rec=1) (fbase(i), i=1,80)

    write(90, 98, rec=1) '/', uyr, umo, '/'
    read(90, 99, rec=1) fdir
    do i = 1, 8
        if ( fdir(i) == ' ' ) fdir(i) = '0'
    end do

    write(90, 94, rec=1) uyr, umo, uda, uhr
    read(90, 95, rec=1) ftime
    do i = 1, 10
        if ( ftime(i) == ' ' ) ftime(i) = '0'
    end do

    write(90, 97, rec=1) '.lr_budi'
    read (90, 92, rec=1) (fsubs(i), i=1,8)
    c = 0
    do i = 1, 80
        if ( (fbase(i) == ' ') .and. (c == 0) ) c = i-1
    end do

    write(90, 92, rec=1) (fbase(i),i=1,c),(fdir(i),i=1,8), &
        (ftime(i), i=1,10), (fsubs(i), i=1,8)

    read(90, 93, rec=1) name

    close(90)

    return

```